PRÁCTICA 7: MRTG

Autores: **Rogelio Montañana**
Juan Manuel Orduña
Francisco Soriano

OBJETIVO Y DESCRIPCIÓN GENERAL.

Multi Router Traffic Grapher (MRTG) es una herramienta diseñada inicialmente para monitorizar la carga de tráfico en enlaces de una red. El MRTG genera páginas HTML que contienen imágenes dinámicas PNG representando el valor a lo largo del tiempo de la variable monitorizada, por ejemplo tráfico en las interfaces de un router o un conmutador, tanto en entrada como en salida. No obstante, MRTG puede utilizarse para monitorizar cualquier variable que se pueda medir. Debido a su utilidad y versatilidad actualmente MRTG se utiliza para monitorizar la evolución de cualquier magnitud durante largos períodos de tiempo (ver por ejemplo http://mrtg.uv.es). El software es de dominio público y puede descargarse de http://www.mrtg.org/ donde también se puede encontrar información detallada sobre su uso.

MRTG se basa en obtener información de los dispositivos a monitorizar bien mediante SNMP (Simple Network Management Protocol) si estos dispositivos disponen de agente SNMP, o bien mediante scripts de usuario. Actualmente existen tres versiones del protocolo SNMP. En esta práctica vamos a monitorizar dispositivos de red (routers y conmutadores) que tienen un agente SNMP versión 1.

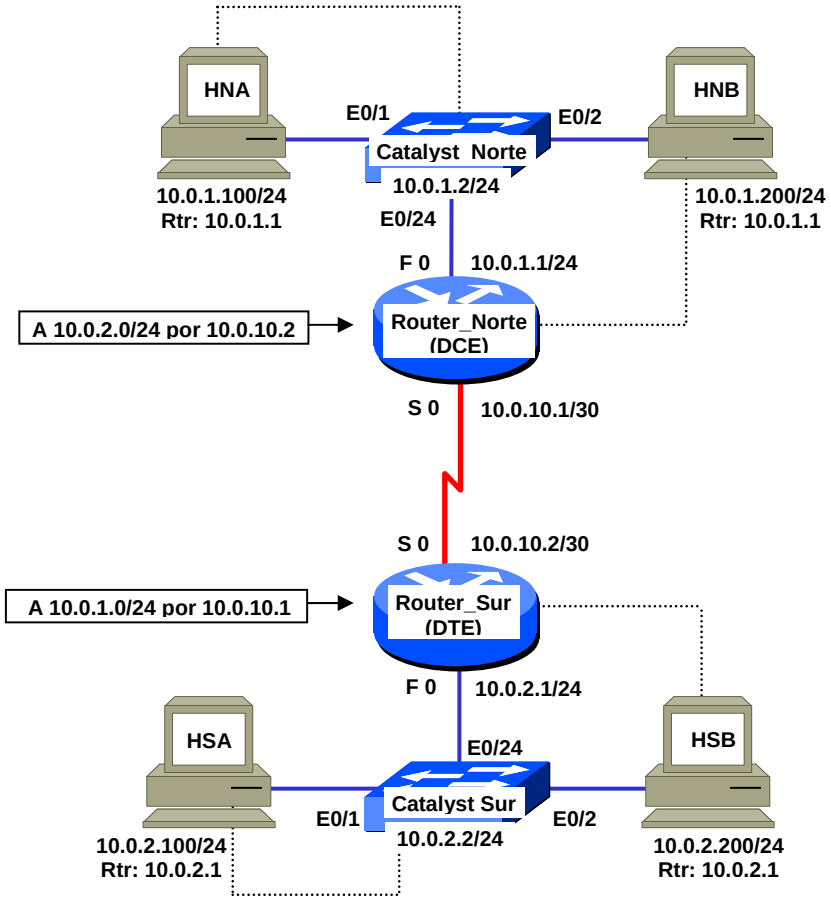


Figura 1. Esquema de la maqueta sobre la que se desarrolla la práctica

Atención: los routers que se utilizan en la práctica no tienen ninguna configuración grabada. Esto puede causar problemas si al arrancarlos está conectada la interfaz serie. Por tanto el cable serie del router deberá dejarse desconectado y conectarse al principio del paso 3, cuando los routers ya tienen una configuración grabada.

Para el desarrollo de esta práctica utilizaremos la misma maqueta de la práctica 6 (ACLs) con unos pequeños cambios. Esta maqueta está formada por dos routers, Router_Norte y Router_Sur, cada uno con al menos una interfaz LAN y una WAN . Los routers están conectados entre sí por la interfaz WAN y en la interfaz LAN tienen conectados sendos conmutadores Catalyst 1924 (Catalyst_Norte y Catalyst_Sur), cada uno de los cuales conecta a su vez dos hosts que denominamos host A (HNA y HSA) y host B (HNB y HSB). La única diferencia respecto a la práctica 6 es que en este caso los conmutadores no tienen la configuración por defecto, sino que se les asignará una dirección IP y un router por defecto; esto es lo mínimo necesario para que el programa MRTG pueda comunicar con los conmutadores y solicitarles la información que se quiere monitorizar. Así, cada LAN utiliza cuatro direcciones IP que corresponden al router, el conmutador y los dos hosts. Las direcciones correspondientes se muestran en la Figura 1. Los hosts se deben conectar a los puertos 1 y 2 del conmutador y la interfaz LAN del router al puerto 24.

El número de maquetas utilizadas en cada sesión de prácticas dependerá del número de alumnos. El número ideal es de cuatro a seis alumnos por maqueta. Cada maqueta funciona de forma independiente del resto durante todo el desarrollo de la práctica, pero dentro de cada maqueta es conveniente que se desarrolle la práctica de forma sincronizada.

Paso 1. Verificación de conexiones y encendido de los equipos

Antes de realizar las conexiones de la maqueta los alumnos deben conectar los ordenadores a la red de la Universidad (rosetas de la pared) y arrrancar los ordenadores con el sistema operativo ‘linux redes’. Una vez arrancado el sistema operativo entrarán con el usuario root y la password que indique el profesor, comprobarán mediante el comando **`'ifconfig'`** que han obtenido una dirección de la red de la Universidad (147.156.*.*) y se bajarán el guión de la práctica de la página web de la asignatura. Es conveniente disponer del guión de forma local durante la práctica para facilitar el desarrollo de la misma.

Una vez obtenido el guión desconectarán los ordenadores de la pared y los conectarán al conmutador según el esquema de la figura 1 (interfaces 1 y 2). A continuación conectarán el router a la interfaz 24 del conmutador con un latiguillo normal (no cruzado). Después conectarán a los ordenadores la consola del router y el conmutador (cables planos negros). Con todo conectado arrancarán el programa **`'minicom'`** en los ordenadores y a continuación encenderán el router y el conmutador. En ese momento verán aparecer por consola los mensajes de arranque. Si no aparece nada deberán comprobar los cables, los equipos y la configuración del minicom, que debe ser la siguiente:

- Velocidad 9600 bits/s
- 8 bits de datos
- Un bit de parada (8N1)
- Sin paridad
- Control de flujo: ninguno
- Dispositivo de entrada: /dev/ttyS0

(El uso del dispositivo ttyS0 se debe a que estamos utilizando el puerto COM1 del ordenador.)

Paso 2. Configuración de los routers

Aunque en la figura 1 y en las configuraciones que siguen se supone que los routers tienen una interfaz WAN y una LAN, y que estas se denominan ‘Serial 0’ y ‘FastEthernet 0’, respectivamente, las interfaces concretas dependen del modelo de router utilizado en cada caso (por ejemplo algunos modelos de router no tienen ‘FastEthernet 0’ sino ‘Ethernet 0’). También puede ocurrir que algunos routers tengan más de una interfaz LAN o WAN. En ese caso los alumnos utilizarán la interfaz de mayor velocidad en LAN y en WAN y si hubiera más de una interfaz de la misma velocidad usarán la de número más bajo (por ejemplo si hay ‘FastEthernet 0’ y ‘Ethernet 0’ usarán ‘FastEthernet 0’, si hay ‘Serial 0’ y ‘Serial 1’ usarán ‘Serial 0’).

Los routers utilizados no tienen grabada ninguna configuración en la memoria permanente. Esto provoca que al arrancarlos entren en un menú de configuración inicial del que debemos salir para introducir la configuración deseada por medio de comandos. Por tanto cuando aparezca la pregunta:

Would you like to enter the initial configuration dialog?

debemos responder **NO.** A continuación aparecerá la pregunta:

Would you like to termiante autoinstall?

A la cual responderemos **YES**. Al cabo de unos instantes obtenemos el prompt de la interfaz de línea de comandos ('Router>').

Para introducir la configuración del Router Norte hay que teclear la siguiente secuencia de comandos:

```
Router>ENABLE
Router#CONFIGURE TERMINAL
Router(config)#HOSTNAME ROUTER_NORTE
ROUTER_NORTE(config)#NO IP DOMAIN-LOOKUP
ROUTER_NORTE(config)#INT F0
ROUTER_NORTE(config-if)#IP ADDRESS 10.0.1.1 255.255.255.0
ROUTER_NORTE(config-if)#NO SHUTDOWN
ROUTER_NORTE(config-if)#INT S0
ROUTER_NORTE(config-if)#IP ADDRESS 10.0.10.1 255.255.255.252
ROUTER_NORTE(config-if)#NO SHUTDOWN
ROUTER_NORTE(config-if)#CLOCK RATE 125000
ROUTER_NORTE(config-if)#IP ROUTE 0.0.0.0 0.0.0.0 10.0.10.2
ROUTER_NORTE(config-if)# CTRL/Z
ROUTER_NORTE#
```

Y para el router sur:

```
Router>ENABLE
Router#CONFIGURE TERMINAL
Router(config)#HOSTNAME ROUTER_SUR
ROUTER_SUR(config)#NO IP DOMAIN-LOOKUP
ROUTER_SUR(config)#INT F0
ROUTER_SUR(config-if)#IP ADDRESS 10.0.2.1 255.255.255.0
ROUTER_SUR(config-if)#NO SHUTDOWN
ROUTER_SUR(config-if)#INT S0
ROUTER_SUR(config-if)#IP ADDRESS 10.0.10.2 255.255.255.252
ROUTER_SUR(config-if)#NO SHUTDOWN
ROUTER_SUR(config-if)#IP ROUTE 0.0.0.0 0.0.0.0 10.0.10.1
ROUTER_SUR(config-if)# CTRL/Z
ROUTER_SUR#
```

Paso 3. Configuración de los conmutadores

Como se ha indicado anteriormente, es necesario asignarle una IP al conmutador y un router por defecto. Ello se hará de la siguiente forma para el conmutador Norte:

```
>enable
#config
Enter configuration commands, one per line.  End with CNTL/Z
(config)#ip address 10.0.1.2 255.255.255.0
(config)#ip default-gateway 10.0.1.1
(config)#CTRL/Z
```

Y para el conmutador Sur:

```
>enable
#config
Enter configuration commands, one per line.  End with CNTL/Z
(config)#ip address 10.0.2.2 255.255.255.0
(config)#ip default-gateway 10.0.2.1
(config)#CTRL/Z
```

Paso 4. Configuración de los hosts

En este paso los alumnos procederán a configurar los hosts con las direcciones y router por defecto haciendo uso de los comandos **`ifconfig`** y **`route`** según se indica a continuación.

En primer lugar deben asignar la dirección IP que corresponde a cada host, de acuerdo con lo que aparece en la Figura 1. Para ello utilizarán el comando:

```
ifconfig eth0 inet dirección_IP netmask máscara
```

Para comprobar que la asignación se ha efectuado correctamente ejecutarán el comando:

```
ifconfig eth0
```

Una vez definida la dirección IP asignaran a los hosts la ruta por defecto mediante el comando:

```
route add default gw dirección_IP
```

poniendo en el campo ***dirección_IP*** la de la interfaz LAN del router que se conecta al mismo conmutador que ese host. Para comprobar que la definición se ha hecho correctamente utilizarán a continuación el comando:

```
route –n
```

El host debe tener ahora dos ó tres rutas definidas que corresponden a la ruta de su propia LAN, la ruta por defecto que acabamos de definir y posiblemente la ruta loopback, que puede aparecer o no dependiendo de la versión de Linux. Por ejemplo en el caso de HNA debe aparecer algo similar a lo siguiente:

```
> route -n
Routing tables
Destination     Gateway         Genmask         Flags   Metric  Ref    Use  Iface

10.0.1.0        0.0.0.0         255.255.255.0   U       0       0      0    eth0
0.0.0.0         10.0.1.1        0.0.0.0         UG      0       0      0    eth0
```

Las rutas definidas mediante el comando **'route add'** se van añadiendo a la lista existente. Por tanto si nos equivocamos deberemos borrar la ruta incorrecta mediante el comando:

```
route del –net 0.0.0.0 gw dirección_IP
```

Para evitar los problemas que puede causar la resolución inversa de las direcciones que intenta hacer por defecto el MRTG y otros programas debemos ahora cambiar de nombre el fichero `resolv.conf` en el directorio `/etc` mediante el comando:

```
mv /etc/resolv.conf /etc/resolv.conf.old
```

Si el fichero `resolv.conf` no existiera en el directorio /etc/ el cambio de nombre daría un error, pero entonces la resolución inversa no nos daría problemas así que no debemos preocuparnos.

Una vez configurados los routers, los conmutadores y los hosts comprobaremos que todo es correcto probando a hacer 'ping' desde los hosts a los cuatro dispositivos de la maqueta (los dos routers y los dos conmutadores). Si el ping funciona consideraremos que la configuración es correcta, si no pediremos ayuda al profesor.

Paso 5. Protocolo SNMP en los conmutadores y routers

Ahora debemos autorizar el uso del protocolo SNMP en los dispositivos. El acceso SNMP se controla por medio de las denominadas **comunidades**, que se identifican por un nombre o cadena de caracteres. Los dispositivos que pertenecen a una misma comunidad pueden intercambiar mensajes SNMP entre sí. El acceso puede ser en solo lectura (RO o Read Only) o en lectura/escritura (RW o Read Write). En modo lectura solo es posible consultar información (contadores de bytes transmitidos por las interfaces o estado de estas por ejemplo) mientras que en modo escritura es posible realizar acciones que modifican el funcionamiento de la red, como activar o desactivar interfaces, borrar contadores, reiniciar equipos, etc. Nosotros utilizaremos acceso RO ya que es lo único que MRTG necesita.

Muchos equipos ya tienen en la configuración por defecto una comunidad con el nombre **'public'** (en minúsculas) para acceso de solo lectura. Este es el caso por ejemplo de los Catalyst 1924, por lo que sin modificar la configuración se les pueden enviar comandos SNMP. Para comprobarlo utilizaremos el comando **'snmpwalk'**. El comando snmpwalk invoca un programa que envía mensajes SNMP (concretamente peticiones 'GET NEXT REQUEST') a un agente SNMP (en nuestro caso el conmutador). Por ejemplo si en una ventana de shell tecleamos:

```
snmpwalk –v 1 –c public 10.0.1.2 interfaces
```

En caso de que tengamos activado el cortafuegos de Linux obtendremos la siguiente respuesta:

```
Timeout: No Response from 10.0.1.2
```

Esto se debe a que el protocolo SNMP se basa en el intercambio de datagramas UDP, que normalmente están filtrados por el cortafuegos de Linux, por lo que para que funcione desactivaremos el cortafuegos utilizando el comando:

```
service iptables stop
```

Al ejecutar el comando **snmpwalk** con el cortafuegos ya desactivado debemos obtener como respuesta la información SNMP relacionada con el grupo MIB-II (Management Information Base II) 'interfaces' del Catalyst Norte (contadores, estado de las interfaces, direcciones MAC, etc.). Esto nos permite comprobar que tenemos acceso SNMP a través de la comunidad 'public' a los conmutadores.

En el comando anterior la opción **'-v 1'** indica que se utiliza la versión 1 de SNMP. La opción **'-c public'** sirve para indicar la comunidad SNMP a la que se quiere acceder. La palabra clave **'interfaces'** indica el grupo de variables MIB-II que se quiere obtener. Otros grupos posibles son por ejemplo **'system'**, **'ip'**, **'snmp'**, etc. Si no se pone este último argumento muestra todas las variables de todos los grupos de MIB-II, lo cual genera un listado bastante largo, difícilmente tratable por consola.

El comando **snmpwalk** también nos permite consultar subgrupos de MIBs. Por ejemplo el siguiente comando nos mostrará las direcciones MAC de las interfaces del Catalyst Norte (ojo a las mayúsculas y minúsculas):

```
snmpwalk –v 1 –c public 10.0.1.2 interfaces.ifTable.ifEntry.ifPhysAddress
```

Incluso es posible especificar una interfaz en particular, por ejemplo si tecleamos:

```
snmpwalk –v 1 –c public 10.0.1.2 interfaces.ifTable.ifEntry.ifPhysAddress.2
```

obtendremos la dirección MAC (dirección física) de la interfaz 2 del Catalyst Norte (en el 1924 la interfaz 2 de SNMP corresponde con la Ethernet 0/2).

Como es evidente las variables MIB-II de SNMP tienen una estructura jerárquica con múltiples niveles, lo cual puede dar lugar a nombres bastante largos a la hora de especificar una variable concreta completamente cualificada, como en el caso anterior. Para evitarlo existe una notación numérica equivalente, que también puede utilizarse con el comando **snmpwalk**. Por ejemplo los siguientes comandos son equivalentes, respectivamente, a los dos comandos anteriores:

```
Snmpwalk –v 1 –c public 10.0.1.2 .1.3.6.1.2.1.2.2.1.6
```

```
Snmpwalk –v 1 –c public 10.0.1.2 .1.3.6.1.2.1.2.2.1.6.2
```

Si intentamos acceder a los routers mediante el comando **'snmpwalk'** anterior veremos que no recibimos respuesta. Esto se debe a que los routers no tienen definida por defecto ninguna comunidad SNMP, ni siquiera la **'public'**. Para definirla es necesario entrar en modo Configure y teclear el siguiente comando:

```
Router_Norte(config)#SNMp-server COMmunity public RO
```

En este comando anterior la palabra **'public'** debe teclearse completa y en minúsculas. Este comando hay que introducirlo en la configuración de ambos routers.

Una vez definida la comunidad public ejecutaremos de nuevo el comando **'snmpwalk'** anterior y ya debemos obtener la información de interfaces correspondiente a los routers.

Llegados a este punto, podemos ejecutar el software MRTG desde cualquier host y monitorizar desde él el tráfico de cualquier router o conmutador, que es lo que haremos a continuación.

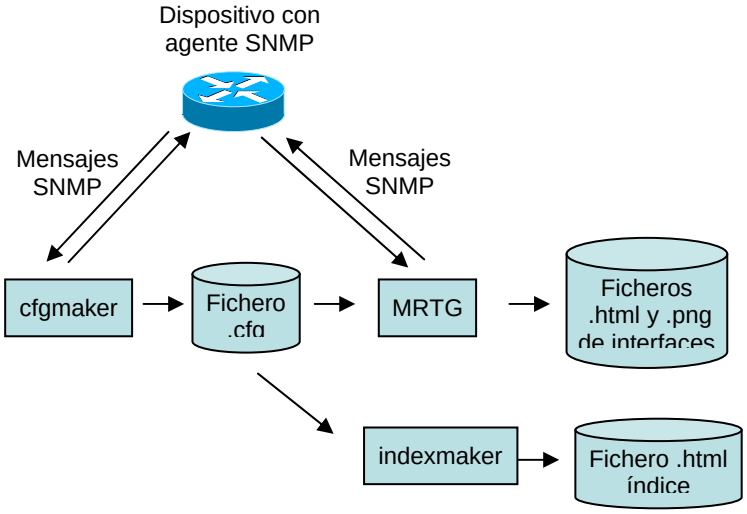
Explicación general del funcionamiento de MRTG y sus utilidades asociadas.

El programa MRTG se basa en un fichero de configuración (normalmente tipo .cfg) para saber que variables MIB debe representar en las gráficas. Ese fichero se puede construir manualmente siguiendo la sintaxis que utiliza MRTG, pero como la cantidad de variables suele ser bastante grande la sintaxis del fichero resultante es tremendamente repetitiva. Por ello existe una utilidad denominada *cfgmaker* para facilitar la elaboración del fichero de configuración, que permite automatizar en buena medida el proceso. La utilidad *cfgmaker* explora mediante mensajes SNMP el dispositivo a monitorizar para averiguar que interfaces tiene y cuáles de ellas están operativas. A continuación *cfgmaker* genera un fichero de configuración adecuado para representar gráficamente el tráfico a lo largo del tiempo en las interfaces que estaban operativas en ese momento. En el fichero de configuración se incluye también la información relativa a las interfaces que no están operativas, pero estas aparecen comentadas dando así la posibilidad al usuario de incorporarlas más adelante si lo desea, descomentando las líneas correspondientes en el fichero de configuración.

Por ejemplo, en nuestro caso los conmutadores Catalyst 1924 tienen 27 interfaces, de las cuales solo 3 están conectadas y el resto están en 'shutdown'. Cuando la utilidad **'cfgmaker'** crea el fichero .cfg incluye el código para monitorizar todas las interfaces, pero deja comentadas las que no se están utilizando, con lo que se evita que MRTG genere un montón de gráficas inútiles.

La información que genera MRTG se estructura en varios ficheros que corresponden a las gráficas .png y las páginas html que las contienen. Puesto que el número de ficheros generados puede ser bastante grande (varios por interfaz monitorizada) es conveniente tener preparado un directorio en el cual se guarden todos los que corresponden a cada dispositivo.

Para facilitar la consulta de las páginas html generadas por el MRTG es conveniente crear una página índice de cada dispositivo que permita acceder a las páginas de las interfaces. Para crear esa estructura de forma automática existe otra utilidad, denominada *indexmaker,* que permite consolidar en una página de índice la información recopilada por MRTG. Para realizar su trabajo *indexmaker* se basa en la información del fichero .cfg creado por *cfgmaker*. El esquema siguiente muestra cómo interactúan las tres herramientas:



Las tres utilidades, *cfgmaker*, *indexmaker* y *mrtg* tienen ayuda en línea mediante el comando *man*.

En una situación normal las utilidades *cfgmaker* e *indexmaker* se ejecutan una sola vez al inicio del proceso. En cambio la utilidad *mrtg* se ejecuta periódicamente (el intervalo típico suele ser de 5 minutos) a fin de obtener la información de los dispositivos e ir rellenando las gráficas con la información obtenida. Por ello el proceso *mrtg* se suele lanzar en modo 'daemon', de forma que él solo se relance periódicamente..

Los pasos a seguir para poner en marcha la monitorización de MRTG son los siguientes:

1. Preparación del entorno de ejecución de MRTG
2. Generación del fichero de configuración con la utilidad *cfgmaker*.
3. Lanzamiento del programa MRTG en modo 'daemon'
4. Ejecución de la utilidad indexmaker, para crear una página html índice de todas las interfaces del dispositivo.

El paso 1 solo se ha de hacer una vez en el host. Los pasos 2, 3 y 4 se han de repetir para cada dispositivo a monitorizar (en nuestro caso dos routers y dos conmutadores).

Vams a describir a continuación cada uno de esos pasos.

Paso 6. Preparación del entorno de ejecución de MRTG

Para su correcto funcionamiento MRTG requiere poner C como lenguaje por defecto de la shell. Para ello debemos teclear el siguiente comando:

```
export LANG=C
```

Si la opción **`LANG=C`** no se añade correctamente el programa *mrtg* no funcionará. Para comprobar que la hemos añadido correctamente podemos utilizar el comando **`env`**. Este parámetro va asociado al shell que se está usando, por tanto si se cierra una ventana de shell y se abre otra nueva será necesario introducirlo de nuevo.

Todos los ficheros de configuración y datos de esta práctica los pondremos en el directorio **`/root/mrtg/`**. Si el directorio no existe los alumnos deberán crearlo, si existe deberán borrar todo su contenido para que no interfiera con el desarrollo de la sesión.

Puesto que *mrtg* y sus utilidades asociadas *cfgmaker* e *indexmaker* generan gran cantidad de ficheros por dispositivo, posiblemente incluso con nombres coincidentes, vamos a crear un subdirectorio para cada uno, lo cual nos permitirá organizarlos mejor. Por tanto ahora debemos crear los siguientes directorios:

```
/root/mrtg/Sw_N                    //Conmutador Norte
/root/mrtg/Sw_S                    //Conmutador Sur
/root/mrtg/R_N                     //Router Norte
/root/mrtg/R_S                     //Router Sur
```

Paso 7. Generación del fichero de configuración con la utilidad cfgmaker

La sintaxis del comando *cfgmaker* es:

```
cfgmaker [options] [community@router] [options] [community@router] ...
```

donde **'community'** es el nombre de la comunidad que se quiere utilizar. Si no se pone, se asume por defecto que es '**public**'. El parámetro '**router**' indica en realidad la dirección IP del dispositivo del cual se desea obtener la información (aunque se indica 'router' el dispositivo puede ser un router, un conmutador, un host, una impresora o cualquier otro dispositivo que incorpore un agente SNMP).

Un ejemplo de invocación del comando *cfgmaker* en la línea de comandos es el siguiente (los caracteres "\\" no forman parte del comando, sino que se utilizan para indicar que lo que sigue debe escribirse en la misma línea):

```
  cfgmaker --global 'WorkDir:  /root/mrtg/Sw_S' \\
--global 'Options[_]: bits,growright' \\
--global 'RunAsDaemon: Yes' \\
--global 'Interval: 5' \\
--output conf_Sw_S.cfg    public@10.0.2.2
```

En este comando le estamos indicando:

- Que cree los ficheros de gráficos y páginas HTML en el directorio **'/root/mrtg/Sw_S'**, El directorio especificado aquí debe existir antes de invocar a la utilidad *mrtg*, y debe especificarse por su path absoluto (empezando por **'/'**).
- Que genere las gráficas de tráfico en bits por segundo (lo normal es en bytes por segundo) y que el eje de abscisas se incremente hacia la derecha **('Options[_]: bits,growright').**
- Que lance el proceso mrtg en modo daemon, de forma que se relance periódicamente de forma automática sin necesidad de invocarlo nuevamente **('RunAsDaemon: Yes')**. La periodicidad de relanzamiento se especifica en la opción **Interval**.
- Que reactive el mrtg automáticamente cada 5 minutos **('Interval: 5')**. Esta opción actúa en combinación con la anterior.
- Que genere el fichero de configuración resultante con el nombre **'conf_Sw_S.cfg'** (en el directorio actual).
- Que queremos generar el fichero de configuración para la comunidad **'public'** del equipo 10.0.2.1.

Las opciones de configuración que aparecen tras la opción '**--global'** se colocan al principio del fichero de configuración y tienen efecto en todo el fichero creado. Debemos tener en cuenta no obstante que el *cfgmaker* traslada las opciones tal cual las tecleamos al fichero de configuración sin realizar

ninguna verificación sintáctica, por lo que si cometemos un error no nos daremos cuenta hasta que más tarde ejecutemos el MRTG con el fichero .cfg.

Si todo ha ido bien habremos creado un fichero de configuración (en nuestro ejemplo '**conf_Sw_S.cfg**') que contendrá la información necesaria para que el programa MRTG pueda monitorizar las interfaces activas, e ir creando las gráficas correspondientes. Las primeras líneas del fichero mostrarán algo parecido a lo siguiente:

```
# Created by
#  /usr/bin/cfgmaker  --global  'WorkDir:  /root/mrtg/Sw_S'  --global
'Options[_]:  bits,growright'  --global  'RunAsDaemon:  Yes'  --global
'Interval: 5' --output conf_Sw_S.cfg public@10.0.2.2


### Global Config Options

#  for UNIX
# WorkDir: /home/http/mrtg

#  or for NT
# WorkDir: c:\mrtgdata

### Global Defaults

#  to get bits instead of bytes and graphs growing to the right
# Options[_]: growright, bits

EnableIPv6: no
WorkDir: /root/mrtg/Sw_S
Options[_]: bits,growright
RunAsDaemon: Yes
Interval: 5

################################################################################
# System: catadec01
# Description: Cisco Systems Catalyst 1900,V9.00.07      written from
147.156.001.143
# Contact: hostmaster@uv.es
# Location: Decanatos - Baja - Rack Baja
################################################################################


### Interface 1 >> Descr: '1' | Name: '1' | Ip: '' | Eth: '00-50-bd-
86-19-41' ###
### The following interface is commented out because:
### * it is operationally DOWN
#
# Target[10.0.2.2_1]: 1:public@10.0.2.2:
# SetEnv[10.0.2.2_1]: MRTG_INT_IP="" MRTG_INT_DESCR="1"
# MaxBytes[10.0.2.2_1]: 1250000
# Title[10.0.2.2_1]: Traffic Analysis for 1 -- catadec01
# PageTop[10.0.2.2_1]: <H1>Traffic Analysis for 1 -- catadec01</H1>
#  <TABLE>
#    <TR><TD>System:</TD>     <TD>catadec01 in Decanatos - Baja - Rack
Baja</TD></TR>
#    <TR><TD>Maintainer:</TD> <TD>hostmaster@uv.es</TD></TR>
#    <TR><TD>Description:</TD><TD>1  </TD></TR>
#    <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
#    <TR><TD>ifName:</TD>     <TD>1</TD></TR>
#    <TR><TD>Max Speed:</TD>  <TD>10.0 Mbits/s</TD></TR>
```

```
#  </TABLE>


                          . . .
                          . . .
                          . . .


### Interface 27 >> Descr: 'B' | Name: 'B' | Ip: '' | Eth: '00-50-bd-
86-19-5b' ###

Target[10.0.2.2_27]: 27:public@10.0.2.2:
SetEnv[10.0.2.2_27]: MRTG_INT_IP="" MRTG_INT_DESCR="B"
MaxBytes[10.0.2.2_27]: 12500000
Title[10.0.2.2_27]: Traffic Analysis for 27 -- catadec01
PageTop[10.0.2.2_27]: <H1>Traffic Analysis for 27 -- catadec01</H1>
 <TABLE>
   <TR><TD>System:</TD>      <TD>catadec01 in Decanatos - Baja - Rack
Baja</TD></TR>
   <TR><TD>Maintainer:</TD> <TD>hostmaster@uv.es</TD></TR>
   <TR><TD>Description:</TD><TD>B  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD>B</TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>100.0 Mbits/s</TD></TR>
 </TABLE>

                          . . .
                          . . .
                          . . .
```

Las primeras líneas de este fichero repiten el comando exacto de *cfgmaker* que ha creado el fichero. A continuación vienen unas líneas que indican parámetros globales para el resto del fichero (‘EnableIPv6’, ‘WorkDir’, ‘Options[_]’, RunAsDaemon: Yes’ e ‘Interval: 5’)

Después viene una descripción que proporciona el propio dispositivo (mediante líneas comentadas) indicando el fabricante, ubicación y demás datos que el administrador de la red haya introducido. Finalmente, para cada interfaz aparece la información necesaria para que el programa MRTG cree la página web. Si la interfaz estaba activa las líneas correspondientes están activadas. Si por el contrario la interfaz no estaba operativa las líneas se comentan, indicando la causa. En el ejemplo anterior la interfaz 1 estaba inactiva y la 27 estaba activa). Además de las interfaces físicas en el fichero de configuración pueden aparecer entradas que correspondan a interfaces virtuales o a algunos elementos singulares que también pueden interrogarse por SNMP, como la carga de CPU el equipo. Estas entradas dependen del tipo de equipo de que se trate y de cómo esté configurado.

TAREA 1: comando cfgmaker

Haciendo uso de la utilidad *cfgmaker* crear en cada host un fichero de configuración .cfg para cada uno de los cuatro dispositivos. Se deben utilizar las opciones **`--global 'RunAsDaemon: Yes'`** y **`--global 'Interval: 5'`**. Se debe especificar un directorio de trabajo diferente para cada dispositivo.

Una vez ejecutado cada *cfgmaker* comprobar que en efecto se han creado los ficheros cfg de configuración correspondientes y han quedado en sus directorios correctos. Si no es así volver a repetir tarea.

Además de monitorizar el tráfico en cada puerto vamos a monitorizar el uso o la carga de CPU en los routers. Los conmutadores 1924 no permiten monitorizar el uso de CPU (otros modelos superiores sí lo permiten).

Dentro del árbol SMI (Structure of Management Information) las MIB-II se encuentran en la rama .1.3.6.1.2.1 (**iso.org.dod.internet.mgmt.mib-2**). Sin embargo el uso de CPU no es una variable MIB-II estándar, por lo que tenemos que bajar por una rama diferente del árbol, que es la que contiene las MIB propietarias de cisco. La rama que corresponde a las variables que reflejan la carga de CPU es la siguiente:

```
iso (1)
      org (3)
            dod (6)
                  internet (1)
                        private (4)
                              enterprises (1)
                                    cisco (9)
                                          local (2)
                                                lcpu (1)
                                                      busyPer  (56)
                                                      avgBusy1 (57)
                                                      avgBusy2 (58)
```

El significado de estas variables (según la documentación de Cisco) es el siguiente:

- **busyPer:** es el consumo promedio de CPU del router (en %) en los últimos cinco segundos.
- **avgBusy1:** es el consumo promedio (media exponencial %) de CPU del router en el último minuto.
- **avgBusy2:** es el consumo promedio (media exponencial %) de CPU del router en los últimos cinco minutos.

Los OIDs correspondientes son por tanto los siguientes:

busyPer: **1.3.6.1.4.1.9.2.1.56.0**
avgBusy1: **1.3.6.1.4.1.9.2.1.57.0**
avgBusy2: **1.3.6.1.4.1.9.2.1.58.0**

Para buscar OIDs se puede utilizar la herramienta disponible en http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en .

Los valores que devuelven estas tres variables deben ser los mismos que aparecen en la primera línea de consola cuando tecleamos el comando:

```
RN#Show PROCesses cpu
```

Como el MRTG consulta los valores del router cada cinco minutos lo lógico sería representar la variable **avgBusy2 (**uso promedio de los últimos cinco minutos), pero nosotros representaremos también la variable **avgBusy1** (valor promedio del último minuto). La variable **busyPer** no la representaremos ya que los intervalos que maneja son demasiado pequeños para los que maneja el MRTG.

Estas MIBs al no ser estándar hay que incorporarlas manualmente al fichero de configuración generado por la utilidad *cfgmaker*, cosa que haremos añadiendo las líneas que aparecen en el cuadro siguiente (ojo, estas serían las líneas a introducir en el fichero de configuración generado por cfgmaker para el router norte, para el router sur es todo igual salvo por la dirección IP):

```
Target[10.0.1.1]: 1.3.6.1.4.1.9.2.1.57.0&1.3.6.1.4.1.9.2.1.58.0:public@10.0.1.1
MaxBytes[10.0.1.1]: 100
XSize[10.0.1.1]: 600
YSize[10.0.1.1]: 150
Options[10.0.1.1]: growright,gauge
ShortLegend[10.0.1.1]: %
WithPeak[10.0.1.1]: dwmy
YLegend[10.0.1.1]: CPU Utilization
Legend1[10.0.1.1]: CPU Utilization in % (avgBusy1)
Legend2[10.0.1.1]: CPU Utilization in % (avgBusy5)
Legend3[10.0.1.1]: CPU Maximal in % (avgBusy1)
Legend4[10.0.1.1]: CPU Maximal in % (avgBusy5)
LegendI[10.0.1.1]:  avgBusy1 
LegendO[10.0.1.1]:  avgBusy5 
Title[10.0.1.1]: CPU avgBusy1 & avgBusy5
PageTop[10.0.1.1]: <H1>CPU avgBusy1 & avgBusy5 for 10.0.1.1<BR></H1>
 <TABLE>
    <TR><TD>System:</TD><TD>10.0.1.1</TD></TR>
      <TR><TD>Maintainer:</TD><TD>hostmaster@uv.es</TD></TR>
 </TABLE>
```

Para facilitar la introducción de las líneas anteriores en el fichero de configuración los alumnos pueden utilizar la función cortar y pegar sobre el guión de la práctica que salvaron localmente al inicio de la sesión. En caso contrario deberán teclear cuidadosamente las líneas una por una.

Paso 8. Lanzamiento del programa MRTG en modo 'daemon'

Una vez creado el fichero de configuración .cfg invocaremos el programa mrtg utilizando como parámetro el nombre del fichero de configuración. En nuestro ejemplo, suponiendo que ya estamos en el directorio donde se ha creado el fichero de configuración, la invocación sería:

```
mrtg conf_Router_Sur.cfg
```

Cada vez que se ejecuta el programa mrtg intenta borrar unos ficheros de ejecuciones anteriores y renombrar otros. Como la primera vez esos ficheros no existen pueden aparecer unos mensajes de 'warning' que son normales. Incluso en la segunda ejecución pueden aparecer algunos 'warnings'.

TAREA 2: comando mrtg

Invocar en cada host el programa *mrtg* cuatro veces para monitorizar los cuatro dispositivos que contiene nuestra red, utilizando para cada uno el fichero .cfg correspondiente.

Una vez se han lanzado los mrtg de los cuatro diuspositivos veremos aparece en cada directorio los ficheros .html y .png correspondientes a las interfaces monitorizadas. Así por ejemplo, en el directorio **'Router_Sur'** el fichero **'10.0.2.2_1.html'** contendrá la página web con las gráficas para la interfaz 1 del Router Sur.

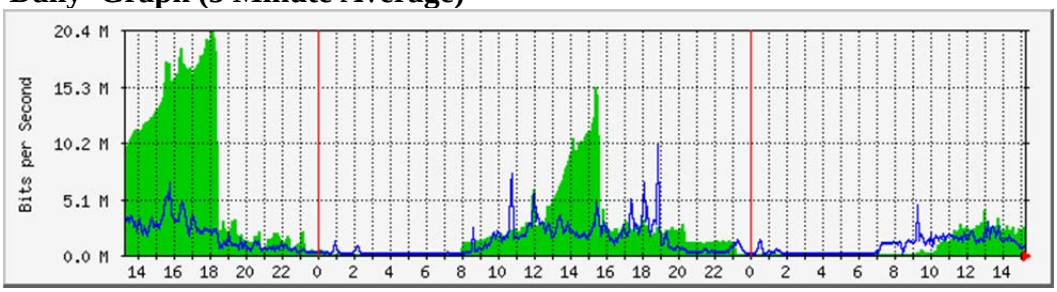
Los ficheros .html obtenidos muestran de forma gráfica el tráfico medio en la interfaz para períodos de tiempo progresivamente mayores de forma similar a la siguiente:

Traffic Analysis for 2/10

System: gigafar Farmacia
Maintainer: hostmaster@uv.es
Interface: 2/10
IP: gigafar.ci.uv.es
Max Speed: 1.0 Gbits/s (ethernetCsmacd)

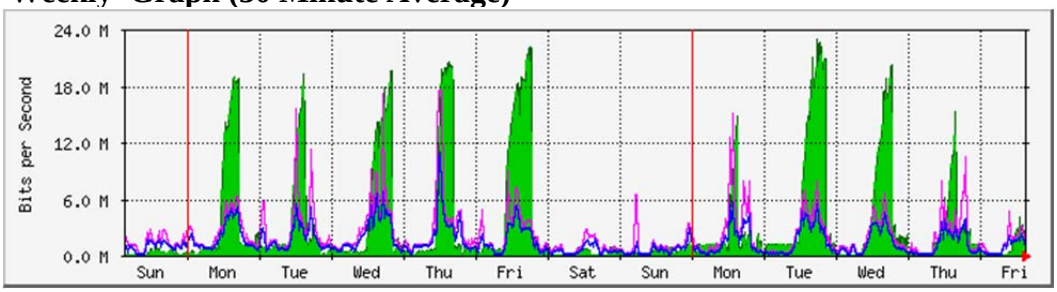
The statistics were last updated **Friday, 21 May 2010 at 15:19**,
at which time '**gigafar.uv.es**' had been up for **17 days, 13:20:23**.

`Daily' Graph (5 Minute Average)



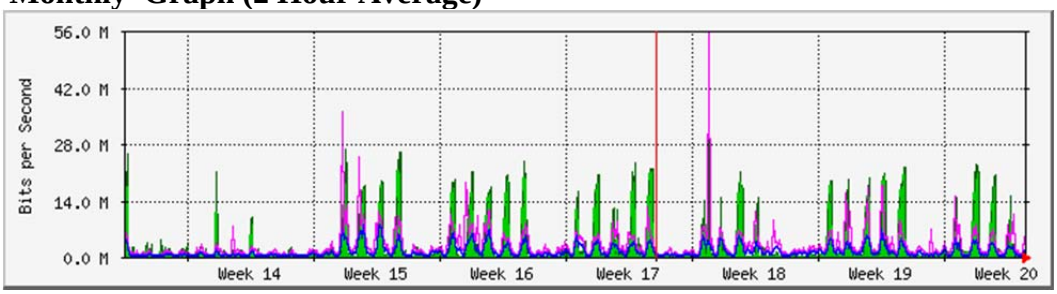
	Max	Average	Current
In	20.2 Mb/s (1.6%)	2922.5 kb/s (0.2%)	2386.2 kb/s (0.2%)
Out	10.0 Mb/s (0.8%)	1152.2 kb/s (0.1%)	532.2 kb/s (0.0%)

`Weekly' Graph (30 Minute Average)



	Max	Average	Current
In	22.9 Mb/s (1.8%)	2998.9 kb/s (0.2%)	2103.6 kb/s (0.2%)
Out	17.4 Mb/s (1.4%)	1352.6 kb/s (0.1%)	931.0 kb/s (0.1%)

`Monthly' Graph (2 Hour Average)



	Max	Average	Current
In	45.0 Mb/s (3.6%)	2325.1 kb/s (0.2%)	1480.6 kb/s (0.1%)
Out	55.3 Mb/s (4.5%)	1190.4 kb/s (0.1%)	1493.2 kb/s (0.1%)

GREEN ### Incoming Traffic in Bits per Second
BLUE ### Outgoing Traffic in Bits per Second
DARK GREEN ### Maximal 5 Minute Incoming Traffic
MAGENTA ### Maximal 5 Minute Outgoing Traffic

MRTG MULTI ROUTER TRAFFIC GRAPHER

2.16.2

Tobias Oetiker <tobi@oetiker.ch>
and Dave Rand <dlr@bungi.com>

En cada gráfica se representan 2 magnitudes codificadas en verde y azul correspondientes al tráfico de entrada y de salida, medidos en bits por segundo. Se pueden escribir plantillas para cada dispositivo de forma que MRTG obtenga cualquier tipo de dato presente en la MIB que utilice el dispositivo. El problema es que no todos los dispositivos usan las mismas MIB, y por tanto no existen plantillas universales que sirvan para todos los dispositivos. En la página web http://people.ee.ethz.ch/~oetiker/webtools/mrtg/ hay un repositorio de plantillas para diversos equipos de diversos fabricantes (bajo el epígrafe **Somix MRTG Configuration Template Archive**) . Estas plantillas pueden ser invocadas en la orden **`cfgmaker.`**

El período de renovación de los datos de MRTG (especificado en la opción **`'Interval'`**) es normalmente de 5 minutos, que es el mínimo posible. MRTG genera una página como esta para cada uno de los puertos activos del dispositivo monitorizado, si el dispositivo es un conmutador de 48 puertos y están todos activos, MRTG generará 48 páginas.

En esta práctica vamos a realizar, condensada en 2,5 horas, la monitorización que normalmente haría un administrador de red durante días, meses o años. Por tanto, las gráficas semanales, mensuales y anuales no nos serán de utilidad.

Si utilizamos la opción **`'RunAsDaemon: Yes'`**, como hemos hecho en este caso, el proceso mrtg lanzado se mantendrá en estado latente y se relanzará periódicamente según lo indicado en la opción **`'Interval'`** para recuperar la información de SNMP necesaria y redibujar las gráficas, Ahora bien, el fichero .cfg solo se lee la primera vez, en caso de que hagamos alguna modificación (por ejemplo cambiar el tiempo de **`'Interval'`**) debemos parar y rearrancar el *mrtg* para que los cambios entren en funcionamiento. El proceso a seguir en ese caso es el siguiente:

1. Averiguar el número de proceso que corresponde al *mrtg* utilizando el comando **`'ps'`**:

```
Host# ps –A|grep mrtg
8649 ?        00:00:00 mrtg
Host#
```

2. Utilizar el commando **`'kill'`** para eliminarlo:

```
Host# kill 8649
Host#
```

3. Relanzar el mrtg:

```
Host# mrtg conf_Router_Sur.cfg
Host#
```

El sistema nos da un mensaje de errror si intentamos lanzar un *mrtg* contra un dispositivo que ya tiene en marcha un proceso *mrtg* en modo Daemon.

Como estamos monitorizando cuatro dispositivos desde un mismo host tendremos cuatro procesos mrtg en modo Daemon que veremos aparecer en la lista del **`'ps'`**. En ese caso no es fácil averiguar a qué equipo en concreto corresponde cada proceso, por lo que si hacemos cambios en uno de ellos probablemente tendremos que eliminar los cuatro y volver a lanzarlos uno por uno.

Paso 9. Ejecución de la utilidad indexmaker para crear una página html índice de todas las interfaces del dispositivo.

Como hemos comentado anteriormente el MRTG construye una página html con su correspondiente fichero .png para cada interfaz activa, lo cual dificulta el acceso a la información, especialmente cuando hay muchas interfaces. Para facilitar esa tarea existe la utilidad indexmaker que construye una página html que actúa como índice para todas las interfaces de un dispositivo. Tecleando 'man indexmaker' podemos consultar información sobre su uso.

Por ejemplo para construir una página de índice de todas las interfaces activas del Router Sur se podría invocar el siguiente comando:

```
indexmaker --output='/root/mrtg/R_S/interfaces.html'
--title='Router_Sur (10.0.2.1)- Gráficas diarias' conf_R_S.cfg
```

TAREA 3: comando indexmaker

Utilizando el comando indexmaker construir una página html donde aparezcan las gráficas de todas las interfaces activas de los cuatro dispositivos de la maqueta.

Paso 10. Generación de tráfico y prueba de funcionamiento de MRTG

Una vez puesta en marcha la aplicación MRTG, el siguiente paso consistirá en generar tráfico entre los distintos hosts de la maqueta y comprobar cómo ese tráfico aparece en las gráficas MRTG para los cuatro dispositivos (2 routers y 2 conmutadores).

Hay que tener en cuenta que, dado que MRTG toma datos cada 5 minutos, y que para calcular tráfico hay que tomar como mínimo dos muestras, deberemos esperar entre 5 y 10 minutos para apreciar algo. Por eso abordaremos en paralelo las tareas 4.1 y 4.2.

TAREA 4.1: lanzamiento de los pings

(mira también la tarea 4.2 en paralelo a esta tarea)

En cada maqueta lanzaremos dos **`'ping –f'`**:

- Desde HSA (10.0.2.100) hacia HSB (10.0.2.200)
- Desde HNA (10.0.1.100) hacia HNB.(10.0.1.200)

Una vez aparezcan valores en las gráficas(al cabo de 5-10 minutos) observa el caudal actual ('Current') que aparece debajo de las gráficas diarias de las interfaces monitorizadas en el conmutador y el router de tu lado de la maqueta, y explica los valores obtenidos. A partir de estos datos calcula el caudal promedio que genera cada uno de los ping –f que estas ejecutando.

Una vez hecha la prueba anterior, y manteniendo en marcha los ping anteriores, lanza ahora los siguientes **`'ping –f'`**:

- Desde HSB (10.0.2.200) hacia HNA (10.0.1.100)
- Desde HNB (10.0.1.200) hacia HSA.(10.0.2.100)

y con los cuatro pings en marcha deja pasar 10 minutos antes de observar cómo cambia el caudal actual ('Current') en las gráficas diarias de las interfaces. Observa y explica las diferencias y calcula a partir de los nuevos datos el caudal generado en las interfaces serie por cada ping –f. El caudal no es exactamente el mismo que en el caso anterior pues el paquete que se envía a nivel de enlace es diferente en una interfaz Ethernet y en una Serie.

TAREA 4.2: análisis de tráfico SNMP con wireshark

Con el objeto de aprovechar el tiempo de espera que se produce en la tarea 4 hasta que se capturan suficientes muestras vamos a analizar con Wireshark el tráfico SNMP que genera MRTG, realizando las siguientes acciones:

- Arranca el wireshark
- Elige el filtro más apropiado para capturar paquetes SNMP. Para ellos ten en cuenta que vamos a capturar solamente paquetes de SNMP de consultas (no traps).
- Analiza algunos de los paquetes capturados en ambos sentidos y responde a las siguientes preguntas:
    - ¿Sobre qué protocolo de transporte va el tráfico SNMP?
    - ¿Qué puertos utiliza el cliente y el agente (o servidor) SNMP?
    - ¿Qué primitivas utiliza MRTG?
    - ¿Qué OID y que MIBs?
    - ¿Qué versión de SNMP estamos utilizando?
    - ¿Qué comunidad estamos utilizando?
    - ¿Con qué frecuencia se envían los mensajes SNMP?

Para hacer más dinámica la captura asegúrate de activar la casilla de captura en tiempo real. De este modo el Wireshark mostrará los paquetes en el momento de la captura.

Paso 10. Apagado y desconexión de los equipos.

Antes de apagar acuérdate de dejar la configuración del resolv.conf como estaba inicialmente con:

```
mv /etc/resolv.conf.old /etc/resolv.conf
```

A continuación debes dejar vacío el directorio `/root/mrtg`. (de lo contrario se podrían producir interferencias con los siguientes compañeros que realicen esta misma práctica).

Así ya podemos proceder a cerrar ordenadamente las sesiones de los hosts y al apagado ordenado de los mismos. Posteriormente, apagaremos tanto los routers como los conmutadores, y finalmente volveremos a conectar cada uno de los hosts a la red de la universidad.

APÉNDICE I

Instalación del paquete MRTG

En caso de que no esté instalado el paquete MRTG en algún ordenador debe seguirse el siguiente procedimiento para instalarlo:

Procedimiento A:

Ejecutar en una ventana de Shell el siguiente comando:

```
yum install mrtg
```

Procedimiento B:

Descargar de www.mrtg.org el fichero mrtg-2.16.2.tar-gz o pedir fichero al profesor. Ejecutar a continuación lois siguientes comandos en una ventana de shell

```
cd /root/Descargas
gunzip –c mrtg-2.16.2.tar.gz | tar xvf-
cd mrtg-2.16.2
./configure –prefix = /usr/local/mrtg-2
make install
PATH=$PATH:/usr/local/mrtg-2/bin
```